AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended)~~ In a system configured to generate code from XML schemas,~~ ~~a~~A method for determining equivalence of XML schema types, the method being performed by a system configured to generate code from XML schemas, the method comprising:

an act of identifying at least two XML schema types for which equivalence is to be determined, each of the at least two XML schema types having at least one schema component that can be presented differently in equivalent XML schema types;

a step for normalizing each of the identified XML schema types;

a step for determining equivalence of the at least two normalized XML schema types.

2. (Previously Presented) The method as recited in claim 1, wherein the step for determining equivalence includes creating and comparing hash numbers of the at least two normalized XML schema types.

3. (Previously Presented) The method as recited in claim 1, wherein the act of identifying the XML schema types includes identifying XML schema types having the same qname.

4. (Previously Presented) The method as recited in claim 1, wherein the step for normalizing each of the identified XML schema types includes writing the at least one schema component in each of the at least two XML schema types according to a unified format and prior to determining equivalence.

5. (Previously Presented) The method as recited in claim 4, wherein writing the at least one schema component includes altering an order of at least two schema components within a single XML schema type.

6. (Previously Presented) The method as recited in claim 5, wherein altering the order includes placing the at least two schema components into alphabetical order.

7. (Previously Presented) The method as recited in claim 5, wherein prior to altering the order, it is determined that the order of the at least two schema components is discretionary.

8. (Previously Presented) The method as recited in claim 4, wherein the at least one component is a discretionary component that is not explicitly recited in at least one of the XML schema types, and wherein writing the at least one schema component includes writing the at least one schema component for a first time.

9. (Previously Presented) The method as recited in claim 1, further including:

upon determining equivalence, creating a single class that is used interchangeably for each equivalent XML schema type.

10. (Previously Presented) A computer program product comprising one or more computer-readable media having computer-executable instructions for implementing a method for determining equivalence of XML schema types, the method comprising:

an act of identifying at least two XML schema types for which equivalence is to be determined, each of the at least two XML schema types having at least one schema component that can be presented differently in equivalent XML schema types;

a step for normalizing each of the identified XML schema types;

a step for determining equivalence of the at least two normalized XML schema types.

11. (Previously Presented) The computer program product as recited in claim 10, wherein the step for determining equivalence includes creating and comparing hash numbers of the at least two normalized XML schema types.

12. (Previously Presented) The computer program product as recited in claim 10, wherein the step for normalizing each of the identified XML schema types includes writing the at least one schema component in each of the at least two XML schema types according to a unified format and prior to determining equivalence.

13. (Currently Amended) ~~In a system configured to generate code from XML schemas, a~~ A method for determining equivalence of XML schema types in a system configured to generate code from XML schemas, the method comprising:

an act of identifying at least two XML schema types for which equivalence is to be determined, each of the at least two XML schema types having at least one schema component that can be presented differently in equivalent XML schema types;

an act of writing the at least one schema component in each of at least two XML schema types according to a custom format resulting in corresponding at least two normalized XML schema types;

an act of comparing the at least two normalized XML schema types;

an act of generating a hash number for each of the at least two normalized XML schema types; and

an act of determining equivalence of the at least two normalized XML schema types when the hash number for each of the at least two normalized XML schema types are the same.

14. (Previously Presented) The method as recited in claim 13, wherein the act of identifying the XML schema types includes identifying XML schema types having the same qname.

15. (Previously Presented) The method as recited in claim 13, wherein writing the at least one schema component includes rewriting an existing schema component into a new format.

16. (Previously Presented) The method as recited in claim 13, wherein writing the at least one schema component includes writing a discretionary component into at least one of the XML schema types.

17. (Previously Presented) The method as recited in claim 13, wherein writing the at least one schema component includes altering an order of at least two schema components within a single XML schema type.

18. (Previously Presented) The method as recited in claim 17, wherein altering the order includes placing the at least two schema components into alphabetical order.

19. (Previously Presented) The method as recited in claim 17, wherein prior to altering the order, it is determined that the order of the at least two schema components is discretionary.

20. (Previously Presented) The method as recited in claim 13, further including:

upon determining equivalence, creating a single class that is used interchangeably for each equivalent XML schema type.

21. (Previously Presented) The method as recited in claim 13, wherein the at least one component is a schema particle definition.

22. (Previously Presented) The method as recited in claim 13, wherein the at least one component is a schema attribute.

23. (Previously Presented) The method as recited in claim 13, wherein the at least one component is at least one of a child and a sub-child of a named type.

24. (Previously Presented) A computer program product comprising one or more computer-readable media having computer-executable instructions for implementing a method for determining equivalence of XML schema types, the method comprising:

an act of identifying at least two XML schema types for which equivalence is to be determined, each of the at least two XML schema types having at least one schema component that can be presented differently in equivalent XML schema types;

an act of writing the at least one schema component in each of at least two XML schema types according to a custom format resulting in corresponding at least two normalized schema types;

an act of comparing the at least two normalized XML schema types;

an act of generating a hash number for each of the at least two normalized XML schema types; and

an act of determining equivalence of the at least two normalized XML schema types when the hash number for each of the at least two normalized XML schema types are the same.

25. (Previously Presented) The computer program product as recited in claim 24, wherein, wherein the act of identifying the XML schema types includes identifying XML schema types having the same qname.

26. (Previously Presented) The computer program product as recited in claim 24, wherein writing the at least one schema component includes rewriting an existing schema component into a new format.

27. (Previously Presented) The computer program product as recited in claim 24, wherein writing the at least one schema component includes writing a discretionary component into at least one of the XML schema types.

28. (Previously Presented) The computer program product as recited in claim 24, wherein writing the at least one schema component includes altering an order of at least two schema components within a single XML schema type.

29. (Previously Presented) The computer program product as recited in claim 28, wherein altering the order includes placing the at least two schema components into alphabetical order.

30. (Previously Presented) The computer program product as recited in claim 28, wherein prior to altering the order, it is determined that the order of the at least two schema components is discretionary.

31. (Previously Presented) The computer program product as recited in claim 24, wherein the method further includes creating a single class that is used interchangeably for each equivalent XML schema type, upon determining equivalence.